

Open Research Online

The Open University's repository of research publications and other research outputs

Proprietary software tools as learning aids

Conference or Workshop Item

How to cite:

Jones, Allan (2007). Proprietary software tools as learning aids. In: AACE Ed-Media 2007, 25-29 Jun s 2007, Vancouver, Canada.

For guidance on citations see [FAQs](#).

© [\[not recorded\]](#)

Version: [\[not recorded\]](#)

Link(s) to article on publisher's website:

<http://www.aace.org/conf/edmedia/sessions/index.cfm/fuseaction/PaperDetails?CFID=3866240&CFTOKEN=24756879&presen>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Proprietary Software Tools as Learning Aids

(Presented at the AACE Ed-Media 2007 conference, Vancouver, 25-29 June 2007)

Allan Jones

Department of Information and Communication Technologies

Faculty of Technology

Open University

Milton Keynes, MK7 6AA, UK

a.jones@open.ac.uk

Abstract: Proprietary software tools, though not designed for educational use, have considerable educational potential. This paper describes, as case studies, the use of proprietary graphics- and audio-editing tools in two distance-taught courses produced by the Department of Information and Communication Technologies at the UK Open University – Europe's largest distance teaching institution. The paper examines the potential advantages and disadvantages of this type of software relative to conventional educational software in terms of constructivist learning theory, and describes how students use the software in the courses concerned. The paper concludes by suggesting further developments of this approach.

Introduction

This paper concerns the use of proprietary software tools in an educational context. 'Proprietary software tools' is taken to mean software that is of commercial origin, and that is designed to modify data files in various ways, under the control of the user. Commercial graphics- and audio-editing packages are prime examples. This kind of software, though not designed for educational use, has considerable educational potential, as the two case studies discussed below show.

Conventionally, software that has been created for educational contexts is generally used for demonstration, simulation or assessment. Even software that has not been created for educational contexts, such as spreadsheets, when used educationally, tends to be used for similar purposes. For example, Bissell (1994) discusses the use of spreadsheets for modelling exercises. However, proprietary software tools, such as audio- or graphics-editing software, facilitate a different educational approach – one based on learning situated in an authentic context.

Brown *et al.* (1989, p. 32) point out that conventional methods of didactic education 'assume a separation between knowing and doing.' Through the use of proprietary software tools, this separation can be bridged, or even effaced. Students can experience an immediate connection between knowing and doing. Brown *et al.* (1989, p. 33) go on to say that: 'People who use tools actively [...] build an increasingly rich, implicit understanding of the world in which they use the tools and of the tools themselves.' Software tools of the kind described here can be help build the rich understanding referred to by these authors.

For many people, problems and solutions are most tractable when they are situated in a physical environment. Brown *et al.* (1989, p.35) quote a much cited incident reported by Lave (1988) when a dieter needed to measure three quarters of two thirds of a cup of ingredients. The solution avoided calculation in favour of physical manipulation of real entities: two thirds of a cup of ingredients were emptied on to a table, formed into a disc shape, and sliced appropriately. Whether this was actually educationally preferable to abstract symbol or number manipulation is debatable (Wineburg, 1989), but it shows how problems can sometimes be solved by shifting the cognitive burden from the abstract realm to the physical environment, where objects can be manipulated in familiar, trusted ways.

The interfaces of software tools are often designed to simulate the physicality of hardware devices. Their displays typically have sliders, knobs, displays, meters, and so on. Using a well designed software tool can feel like performing a physical activity, and familiarity with it often leads to the kind of comfortable sensation associated with using a favourite kitchen knife or screwdriver – an equivalent sense of exasperation when a new model appears with a different design.

Constructivist principles underlie much contemporary learning technology. Among the several defining characteristics of constructivism, one is that 'knowledge is built from our interactions with the environment'

(Gibson & Jonassen, 2000, p. 180). Knowledge building, then is an active process involving the learner, the environment, learning aids, and the learner's existing knowledge. If the learning aids include computers, then they become 'a learning partner, rather than a medium for direct instruction' (Oliver & Herrington 2000, p. 179).

Proprietary software tools fit into this constructivist approach readily by virtue of their authenticity; that is, they 'reflect the ways in which knowledge and learning outcomes are to be used in real-life settings beyond the classroom' (Oliver & Herrington, 2000, p. 180). Adobe Audition, one of the software tools discussed in this paper, for instance, has a long history of use in professional local-radio production. As Palinscar (1989) says: 'learning that occurs within a meaningful context has heightened value.'

In addition to the educational benefits summarised above, a software tools-based approach can have other potential benefits:

- The software already exists.
- Students can gain useful skills, sometimes transferable.
- The software might be cheap, or free.

Some of the drawbacks include:

- Software might be ill-suited to educational purposes.
- Software is likely to have many more facilities than the teacher wishes to use, which could be confusing for the student.
- Software might be expensive.

The two pieces of software discussed here represent both the free and the moderately expensive parts of the spectrum. Irfanview is a freeware graphics editor, and Adobe Audition is a moderately expensive audio editor.

Irfanview graphics software

Irfanview is issued to students of the UK Open University course T175 Networked Living. This course is an introduction to information and communication technologies. It is presented twice per year with a typical student enrolment per presentation of around 1400 students. In customary Open University fashion, students study at home using a range of resources created or designed by a team of academic staff. Students' ages, prior knowledge, and experience, cover a wide range.

The course requires approximately 300 hours' of study, over several months. When students are about two-thirds of the way through the course, they study some of the theoretical principles of digital image representation and file compression. Following this theoretical study, students do a few hours of practical activities using the Irfanview graphics editor.

The Irfanview activities essentially investigate the problem of choosing appropriate file formats for certain graphics. Why, for instance, would it be unwise to use JPEG format for (Fig. 1) and (Fig. 2), but sensible for (Fig. 3)? On the other hand, why would it be sensible to use certain types of TIF file, or PNG (portable network graphic) file, for (Fig. 1) and (Fig. 2), but not for (Fig. 3)?



Figure 1: Pen-and-ink sketch

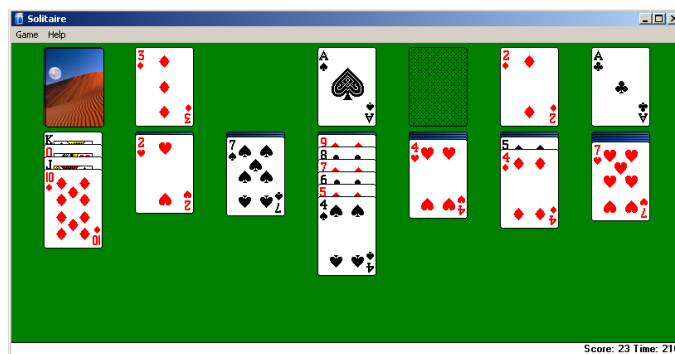


Figure 2: Screen shot of Patience game

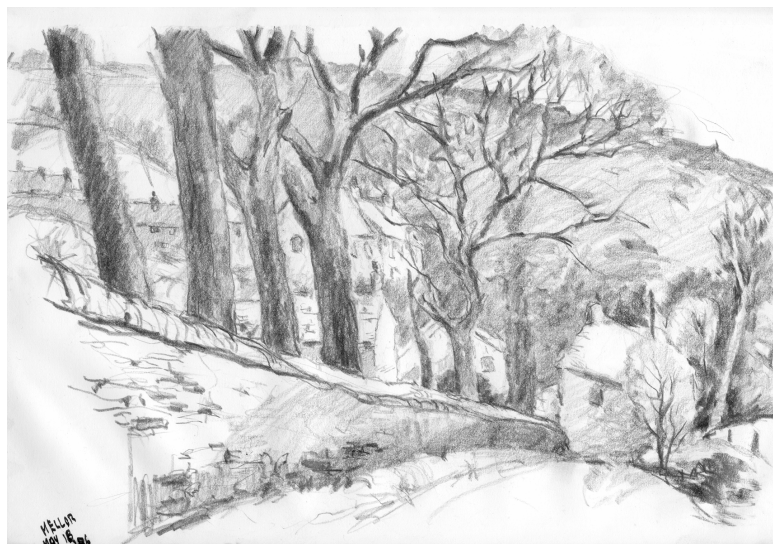


Figure 3: Pencil sketch

The answers to these questions, as students are taught in prior theoretical work, relate to the degree of repetition of pixel values. In (Fig. 1) there are only black and white pixels (no greys), with long sequences of the same value. (Fig. 2) similarly has long sequences of identical pixel values where there are areas of the same colour. It also has some recurring patterns of pixels. This high degree of repetition allows lossless data compression to be

used to reduce file size. Lossless compression enables the original data to be reconstructed perfectly from the compressed version. It is built in to the PNG format, and available as an option in the TIF format. Students use both formats as part of the activities.

However, pictures like (Fig. 3), or photographs of natural scenes, are characterised by a lack of repeated sequences (or patterns) of pixel values. Even areas of the picture that appear to be the same colour are found, on close examination, to consist of pixels with slightly different values. Lossless compression is largely ineffective here, as students discover. Instead, lossy compression can be used. Lossy compression discards data irretrievably, but in ways that are imperceptible, or nearly so, to human perception. JPEG is an example of a format that uses lossy compression.

Irfanview practical work

The practical work with Irfanview gives students an opportunity to test some of these theoretical principles. Students are provided with Windows bitmap files (uncompressed) of (Fig. 1) and (Fig. 3), and some other images. They use these files in a range of activities in which they modify the files, save them into various formats, and check the result visually and in terms of files size.

Many other proprietary graphics editors would serve equally well for these practical activities. However, Irfanview has features that make it well suited for educational use, two of which are singled out here. First, it is freeware. (Permission to distribute the software still has to be sought from its creator, of course.) Second, Irfanview has an Image Properties window which usefully brings together information relating to a picture. (Fig. 4) shows the image properties window for (Fig. 3). (The line numbering down the left-hand side has been added for ease of referencing in this paper.) The activities make extensive use of this window.

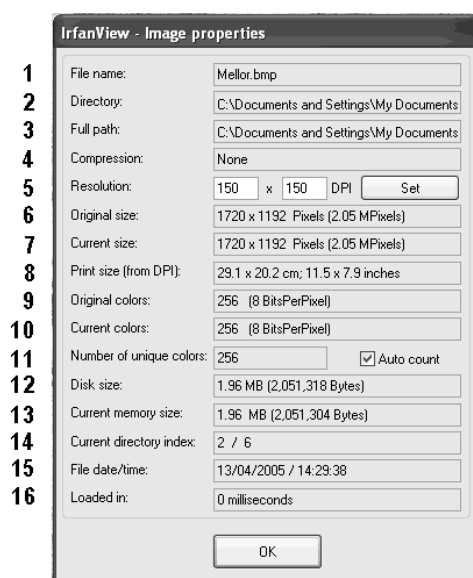


Figure 4: Image properties window

Line 12, 'Disk size', is the size of the file on the hard drive. If the file format uses compression, this is the compressed size. Line 13, 'Current memory size', is the size of the file in the computer's RAM. This is the uncompressed file size. Comparison of lines 12 and 13 thus gives a quick comparison of compressed and uncompressed file sizes. In (Fig. 4), compressed and uncompressed sizes are virtually identical because, as line 4 shows, there is no compression in the file format used.

Line 11 of (Fig. 4) shows the number of colours present in a picture, or, in a greyscale picture, the number of levels of grey. In this case there are 256 shades of grey. Lines 9 and 10 show that 8 bits per pixel are used, which means that the maximum number of different colours that can be represented is 2^8 , or 256.

The number of bits per pixel affects both the number of quantised colour levels available and the file size. The more bits there are per pixel, the larger is the uncompressed file size. What would be the consequence of reducing the number of bits per pixel from eight to four? The answer, as the students find out, is that it halves the file size. However, using only four bits per pixel reduces the number of available levels of grey from 256 (2^8) to 16 (2^4). Is the effect noticeable? Once again, students find out for themselves. The answer is that the result is hardly detectable by eye. Reducing the number of bits per pixel from 8 to 4 is effectively a kind of lossy compression: the file size is reduced by discarding visual information. Students investigate further reductions in the number of grey levels.

Further activities involve saving (Fig. 1) and (Fig. 3), and colour photographs, using formats that have either lossless or lossy compression built in. Students check the quality of the image and the size of the file to see, for instance, whether artefacts have been created. Along the way, students encounter cognitive conflict when they find that using compression on the wrong sort of image can actually increase the file size. The reasons are explained to students.

Student responses to the Irfanview activities tend to stress the importance of doing things for oneself:

‘I found it fascinating to be gently led through some of the principles and see the results for myself. I didn’t understand greyscale or anything about colour representation and am happy to have learnt a little bit about them.’

Many students were, in fact, already familiar with the use of graphics editors, usually through digital photography. (Digital cameras are generally supplied with software that has graphics editing capabilities.) However, these students too were able to benefit from the work:

‘I was somewhat familiar with these concepts but enjoyed working through the materials to consolidate my knowledge.’

Thus prior familiarity with this type of software need be no obstacle, and can give extra ‘intrinsic’ motivation. According to Biggs (2003, p. 62), intrinsic motivation ‘drives deep learning and the best academic work.’

Adobe Audition

The proprietary audio editing package Adobe Audition (version 1) is issued to students of the course TA212 The Technology of Music. In a course on the technology of music, the use of an audio editor hardly needs any justification. However, in addition to using the software to perform audio editing functions, students use it to explore some theoretical ideas.

The Technology of Music course is at a higher academic level than the course which uses Irfanview, and the theoretical concepts are considerably more advanced. Students spend a week working with Audition, and investigate several effects in acoustics and psychoacoustics. Two are described below.

It is fairly well known that the ear/brain’s response to changes of sound amplitude is nonlinear. Equal absolute changes of amplitude are not perceived as equal changes of loudness. However, changes of amplitude by a constant factor (such as doubling, or halving), to most people, sound like equal steps of loudness change.

(Fig. 5) shows a screen from the Audition software. The light coloured horizontal band across the middle represents 10 seconds of a sound. The depth of the band from top to bottom indicates the amplitude (strictly, the peak-to-peak amplitude). At the left of the display, the light band has its greatest amplitude. Moving to the right, the depth of the band suddenly changes to half its former value, corresponding to a halving of amplitude. There are three further halvings of amplitude as the display is traversed from left to right.

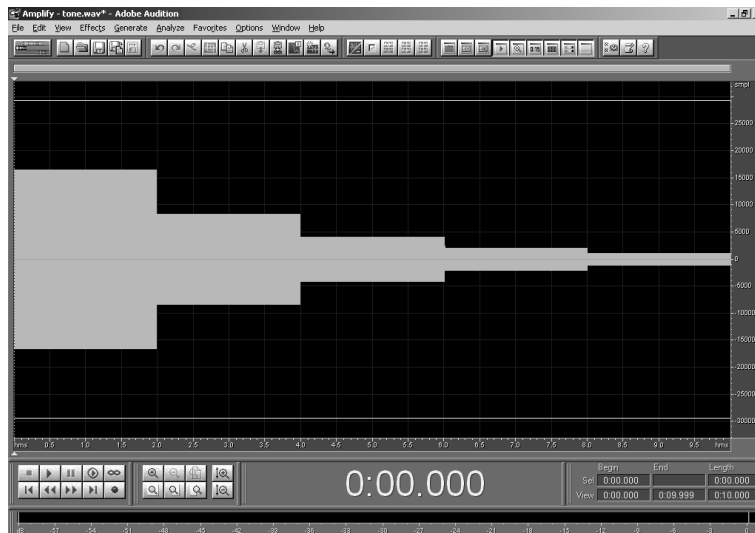


Figure 5: Audition screen showing four halvings of amplitude

The sound represented in (Fig. 5) actually has a sinusoidal waveform, but at this scale the individual cycles of the waveform cannot be discerned. However, by zooming in on any part of the display, the sinusoidal shape becomes visible.

When the waveform is made audible (by clicking on a button), a cursor moves across the display from left to right, its position corresponding to the part of the waveform that is being played at that instant. The student hears the changes of amplitude, and sees what these correspond to. Most people judge that the loudness is decreasing by equal steps.

The important point of this exercise from an educational point of view is that the students construct the waveform in (Fig. 5) themselves. It would have been very easy to have supplied students with a pre-recorded example, or a pre-prepared audio file incorporating the amplitude changes. Instead, the students generate for themselves ten seconds of unchanging waveform, and then halve the amplitude of part of it, then choose part of the halved section and halve its amplitude, and so on. The repeated act of halving the amplitude brings home the idea that there is a consistent change in all the steps in (Fig. 5), even though the steps are of different absolute heights. In the process of constructing the file, students see how easy it would be to choose other amplitude changes than halvings, and are at liberty to pursue this. Students also, incidentally, see the amplitude change specified in decibels (which they have met in theoretical work).

The second example from the Audition activities is a demonstration of the difference between tuning a chord according to frequency ratios derived from the harmonic series and ratios specified by equal temperament, which is the tuning system generally used on modern instruments with fixed tuning (for example, keyboard instruments, fretted instruments and woodwind instruments). The harmonic series is a naturally occurring series of frequency relationships, generally regarded as providing the origin of our way of subdividing the octave. Pitches in equal temperament deviate from those derived from the harmonic series in a way that makes all musical keys available to a performer without the need to adjust the tuning from one key to another.

One version of a chord of A major consists of the following notes, in order of ascending pitch: A, C#, E, A, E. (Other arrangements of these note names also constitute a chord of A major.) The relative frequencies of the notes in this chord can be specified by taking the lowest note, the first A, to have a frequency of 1. The relative frequencies of this set of notes would then be: 1, 1.25, 1.5, 2, 3.

(Fig. 6) shows this chord being constructed in Audition. In the centre of the screen, the above series of relative frequencies has been entered by the student.

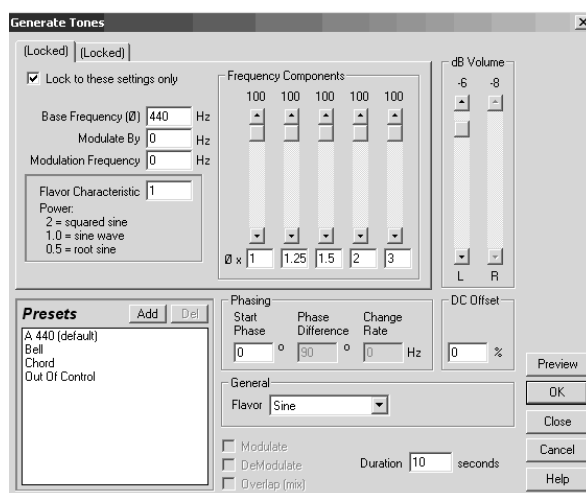


Figure 6: Constructing a chord of A major

Towards the top left of (Fig. 6), the ‘Base frequency’, which is the frequency of the lowest note in the chord, has been set to 440 Hz. This is the frequency of the first A above middle C. Entering a different base frequency, but keeping these relative frequencies, would give a chord based on a different note, but still tuned according to the harmonic series.

Having constructed this chord, and heard the result, students construct the same chord using the relative frequencies of equal temperament, which are (to a close approximation): 1, 1.26, 1.4983, 2, 2.9966.

The difference is immediately apparent visually in Audacity. (Fig. 7) shows that the chord tuned to the harmonic series (labelled ‘pure’) has a steady amplitude, whereas the equally tempered chord has an undulating amplitude caused by beating of waves whose relative frequencies are not quite simple numerical values.

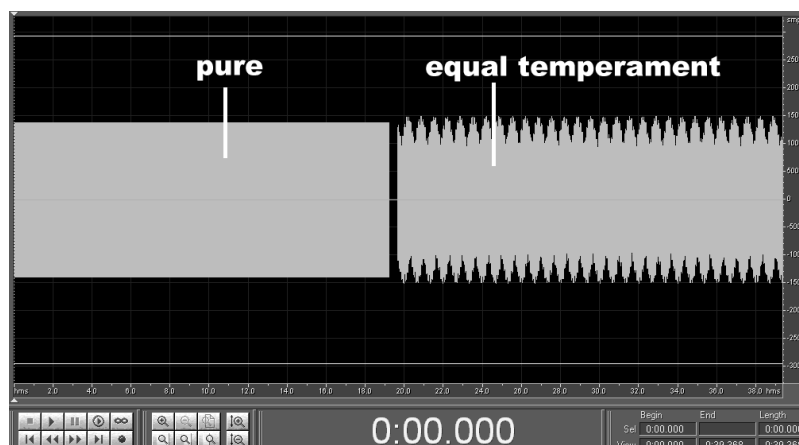


Figure 7: Chord tuned according to ‘pure’ frequency ratios of the harmonic series, followed by same chord in equal temperament

Playing the sound reveals the equally tempered chord to have a somewhat shimmering quality. Students might observe that the equally tempered chord does not sound out of tune, but has a different sound quality from the ‘pure’ version. The concept of ‘out of tune’ is not an easy one to unpick.

Very little of this exercise in temperament comes to the students ‘out of the blue’. By the time they tackle this exercise, students have already read about, and heard demonstrations of, the harmonic series, chords, beating, and equal temperament. The act of entering the relative frequencies brings home the fact that some relative pitches in the equally tempered chord deviate more from the harmonic-series values than others. In creating the sounds for themselves, students can see how easy it would be to experiment further – for instance to see the effect of restoring individual notes in the equally tempered chord to pitches specified by the harmonic series, or

by testing how far the constituent notes can depart from the harmonic-series values before the chord sounds out of tune.

Student responses to the Audition activities have been very favourable. Sixty per cent of the students surveyed in 2004 said this material was 'very useful', and 31% considered it 'fairly useful'. However, it needs to be stressed that the activities with Audition serve a dual purpose – familiarising the students with Audition itself and exploring concepts taught earlier, such as equal temperament.

Conclusions

This paper has made a case for the use of proprietary software tools in education based on:

constructivist learning environment – the learner performs actions situated within the quasi-physical interface of the software

authenticity – the software is genuine for the tasks it performs

The case studies show that this approach can be used successfully. However, as will be clear from the foregoing, successful use of these software packages as educational tools depends on preparation. Students need to be introduced to relevant concepts beforehand, but there must also be scope for surprise and discovery. Students should not simply follow a routine or just be left to their own devices.

In principle, a software-tools approach would appeal to students with a preferred activist learning style, but if the theoretical groundwork is laid, the approach can be equally attractive to students whose preferred styles are theorist or reflective (using the learning styles terminology of Honey & Mumford, 1986).

Only graphics and audio editing software have been discussed here, but other sorts of software offer similar potential. Video editing software is the most obvious example of another software tool that could be adapted to educational use. Other possibilities include audio recording software (audio editors such as Audition often double as recording software) and optical scanner software.

This paper does not claim that a tools-based approach is an educational panacea. The use of software tools does not displace other teaching methods or other types of educational software. It is best viewed as an addition to the armoury of teaching approaches.

References

- Biggs, J. (2003). *Teaching for quality learning at university*. Second edition. Buckingham: Society for Research into Higher Education & Open University Press.
- Bissell, C. (1994). Spreadsheets in the teaching of information engineering. *Engineering Science and Educational Journal*, April, 89–96.
- Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, January/February, 32–42.
- Gibson, C. C., & Jonassen, D. H. (2000). Speaking personally. In Moore, M. G., & Shin, N. (eds), *Speaking Personally about Distance Education*, Readings in Distance Education no. 6. University Park, Pa: American Center for the Study of Distance Education.
- Honey, P. & Mumford, A. (1986). *The manual of learning styles*. Maidenhead: Peter Honey Associates.
- Lave, J. (1998). *Cognition in practice*. Cambridge: Cambridge University Press. 165.
- Oliver R., & Herrington, J. (2000). Using situated learning as a design strategy for web-based learning. In Abbey, B. (ed.), *Instructional and cognitive aspects of web-based education*. Hershey, Pa: Idea Group Publishing.
- Palinscar, A. S. (1989). Less charted waters. *Educational Researcher*, May. 6.

Wineburg, S. (1989). Remembrance of theories past. *Educational Researcher*, May, 9.

Acknowledgement

Drawings in (Fig. 1) and (Fig. 3), by E. Allan Jones (1919–2000), reproduced by kind permission of Dorothy Jones.